

## HLRmon: a role-based grid accounting report web tool

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2008 J. Phys.: Conf. Ser. 119 052012

(<http://iopscience.iop.org/1742-6596/119/5/052012>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

### Download details:

IP Address: 192.99.32.115

This content was downloaded on 18/06/2014 at 03:19

Please note that [terms and conditions apply](#).

# HLRmon: a Role-based Grid Accounting Report Web Tool

## Stefano Dal Pra

INFN-Padova, Padova, Italy

E-mail: [Stefano.DalPra@pd.infn.it](mailto:Stefano.DalPra@pd.infn.it)

## Enrico Fattibene, Giuseppe Misurelli

INFN-CNAF, Bologna, Italy

E-mail: [Enrico.Fattibene@cnafe.infn.it](mailto:Enrico.Fattibene@cnafe.infn.it), [Giuseppe.Misurelli@cnafe.infn.it](mailto:Giuseppe.Misurelli@cnafe.infn.it)

## Federico Pescarmona, Luciano Gaido

INFN-Torino, Torino, Italy

E-mail: [Federico.Pescarmona@to.infn.it](mailto:Federico.Pescarmona@to.infn.it), [Luciano.Gaido@to.infn.it](mailto:Luciano.Gaido@to.infn.it)

**Abstract.** Both Grid users and Grid operators need ways to get CPU usage statistics about jobs executed in a given time period at various different levels, depending on their specific Grid's role and rights. While a Grid user is interested in reports about its own jobs and should not get access to other's data, Site or Virtual Organization (VO) or Regional Operation Centre (ROC) manager would also like to see how resources are used through the Grid in a per Site or per VO basis, or both. The whole set of different reports turns out to be quite large, and various existing tools made to create them tend to better satisfy a single user's category, eventually despite of another.

HLRmon[1] results from our efforts to generate suitable reports for all existing categories and has been designed to serve them within a unified layout. Thanks to its ability to authenticate clients through certificate and related authorization rights, it can a-priori restrict the selectable items range offered to the web user, so that sensitive information can only be provided to specifically enabled people.

Information are gathered by HLRmon from a Home Location Register (HLR) which stores complete accounting data in a per job basis. Depending on the kind of reports that are to be generated, it directly queries the HLR server using an ad-hoc Distributed Grid Accounting System (DGAS[2]) query tool (typically user's level detail info), or a local RDBMS table with daily aggregate information in a per Day, Site, VO basis, thus saving connection delay time and needless load on the HLR server.

## 1. Introduction

In order to provide graphical or tabular reports about resources usage (mainly CPU times) in a Grid environment, several steps and distributed components are involved.

DGAS collects and stores information about any job executed. Among these information the following are of particular interest for us: the Site name where the job got executed, the

belonging Virtual Organization, which user ran it, when it actually started and finished, how much CPU time (with normalization factor) and overall time it took.

Those data are kept into a MySQL database by a HLR which is usually located at a Site level and can eventually collect data coming from near Sites too. A “second level” HLR exists at ROC level, whose purpose is to “mirror” data from lower level HLRs thus offering a centralized source of information for job executed in the whole ROC. Jobs get accounted into the HLR after they terminate execution and are stored with respect to the day of start.

A textual tool is provided within the DGAS middleware which permits local users as well as remote and authorized users to query a HLR data base.

Within this scenario, the further needed step is to provide a grafical easy-to-use reporting tool which retrieves data from the right HLR (the second level HLR may or may not collect all lower level ones) using the textual query tool and presents them in a suitable way to the final user. HLRmon has been designed to perform these steps. Furthermore, it takes in charge to limit the scope of accessible data depending on the particular user’s role.

## 2. HLRmon architecture and features

HLRmon low-level architecture makes use of the following components (figure 1): one or more HLR databases (one of them may be a second level one), as data sources; a binary command line HLR query tool acting as a HLR client, a software module which composes queries to the proper HLR and submits them launching a query tool instance, catching the “raw text” output and converting it in a suitable format (runtime generated data structure or SQL INSERT statements) for the web presentation layer.

In order to provide faster response to the final user, data related to aggregated daily activity of Sites and VOs are nightly retrieved and locally stored in a MySQL database. The “Report” section of the HLRmon tool only provides charts or tabular results from this local database, which collects daily overall number of jobs executed by a VO at a given Site, with total CPU time and Wall time (normalized and not). The displayed table can be ordered by a field of user’s choice; the actual ordering algorithm is performed by a javascript routine which runs at client side. This saves undesirable load on the webserver.

At a web-level, layout has been designed keeping in mind user’s comfort and friendliness. To accomplish this a “persistent customization” mechanism has been implemented, which keeps track of selections individually made by users and maintain them at any new web access from the same user. Here “same user” actually means “a web client with a valid digital certificate from a known, registered and enabled user.”

The “Ranking” section of HLRmon provides statistics about user’s activity: who submitted how many jobs to which VO and Site and how much CPUtime and Walltime were used. The result is displayed in a tabular form which can be eventually exported into an excel file, for local analysis. A “top-ten” subsection is being implemented, to list at a glance most active users respect to number of submitted jobs or respect to CPU usage.

This kind of information is runtime gathered from the second level HLR and may be slower to get ready respect to those available in the “Report” section. An effective cache mechanism is however adopted here too, avoiding the need to perform twice already made queries. This way, “most wanted” pages are often already available because cached from an identical request.

### 2.1. HLRmon and web users

This section explains how users gets recognized, registered and enabled to access information depending on the scope associated with their roles. We address four types of role: “ROC manager”, who can access anything related to any Site and VO; “Site manager”, who can access anything related to his Site; “VO manager”, who can see activity of his VO between any Site. At

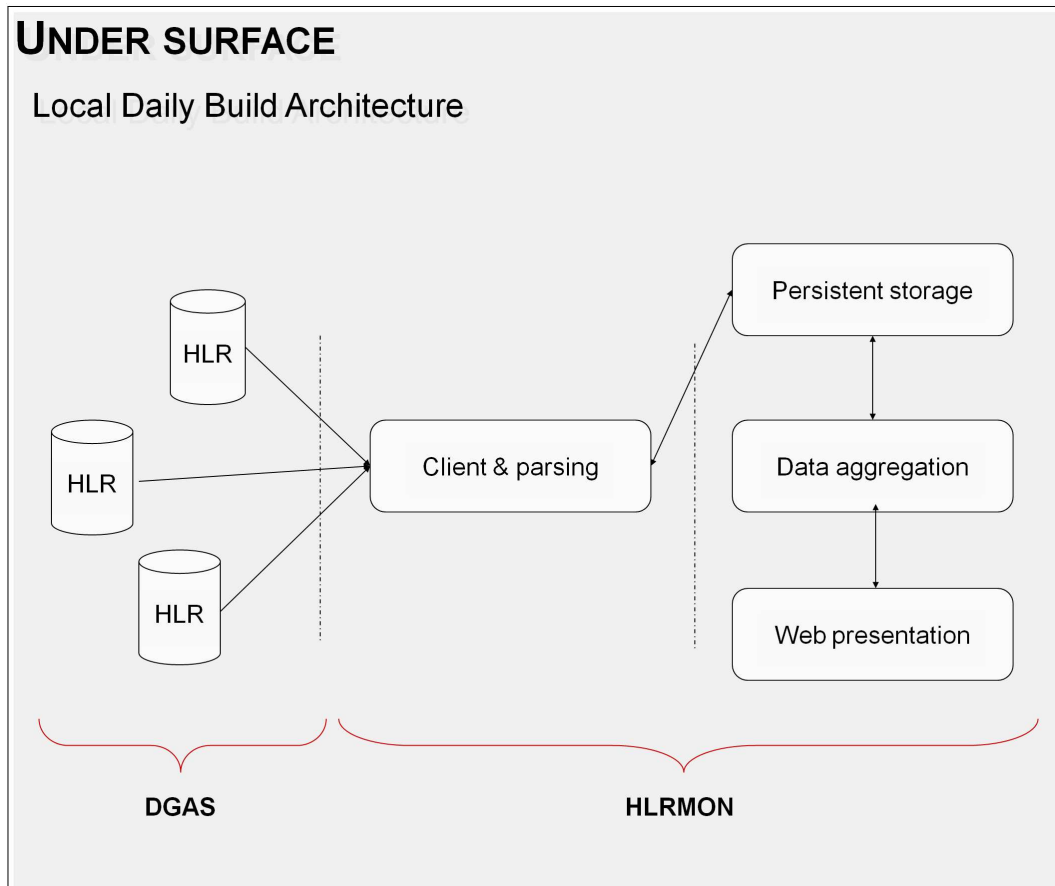


Figure 1. HLRmon architecture

a lowest level there is the “VO user” role. A VO user can only see his own jobs in the “Ranking section”, i.e. jobs he ran at any Site.

HLRmon recognizes users through the Distinguished Name of their digital certificate, which must be loaded into the browser they use. Clients without a valid or acceptable certificate get redirected to a “no-cert” page which explains this.

Clients with a valid and “never-seen” certificate are redirected to a “welcome” page with a link to a “registration” form. There a user can select desired roles (a same user may be enabled for many different roles). After the form submission the new user gets registered into the local database with status “disabled” and an email gets delivered to the HLRmon staff, which evaluates and enable (or refuse!) client requests. Users registered but not yet enabled will be redirected on a “waiting” page informing them that their status has not yet been enabled.

Once enabled the client can access HLRmon pages with granted privileges enabled. He will be first redirected on the “Charts” section. Here a default set of charts will be displayed according to the selections from the menus on the left:

- The desired role among available ones (higher one as default).
- The visible Sites list according to his roles (all pre-selected).
- The visible VOs list according to his roles (all pre-selected).
- The date interval to which restrict the search (last 30 days as default)

Field	Description
YYMMDD	Date
Site	Name of the Site
VOname	Name of the VO
TotalJobs	Total job number
TotalCPU	Total CPU Time [s]
TotalWall	Total Wall Time [s]
NormTotalCPU	Normalized Total CPU Time [s]
NormTotalWall	Normalized Total Wall Time [s]

**Table 1.** Locally stored aggregated data

One or more Site name in the Site list may be red coloured, meaning that no news has been received from that Site during latter few days. A tooltip will inform about how many. This info may be useful to the Site manager who can then investigate whether some kind of accounting problems may be arised at his Site or at a next step in the chain.

An user can customize default selections at will, including/excluding any desired Site/VO and adjusting date interval for subsequent queries. A “search” button is of course provided to re-generate the page according to the new selections, as well as a “reset” one to get back to default. These settings will be remembered for further access; the time interval will be conveniently shifted of the number of elapsed days after previous user access.

## 2.2. Security and Privacy

HLRmon has been designed and engineered to fulfill security and privacy constraints. Users' specific roles are enabled or disabled by system administrators; users can only disable a role on their own.

The whole system is based on a security layer which appends to every single query user's specific permissions, thus avoiding to show data they should not see. The system is furthermore based on digital certificates, which represent the state-of-the-art for remote identification, still allowing anywhere and anytime access to the web application.

Cached graphs and raw data are maintained for a limited time and stored in hash files, thus allowing simple and fast server-side retrieving but causing a nearly impossible reading (user should guess a 64-bit code to access a single file) by remote access.

The system has been tested deeply against various types of attacks including SQL injection, proving safeness and security.

## References

- [1] HLRmon Website. <https://dgas.cnaf.infn.it/hlrmon>.
- [2] DGAS Website. <http://www.to.infn.it/grid/accounting/main.html>.